

## AMENDMENTS TO THE CLAIMS

*Please cancel claims 2, 8, 17, and 23 without prejudice.*

*Please amend the claims as follows:*

1. (Currently amended) A method comprising:  
~~receiving a series of operations, an operand of the series of operations being  
converted from a first format to a second format prior to performing a first  
operation of the series of operations, a result of the series of operations  
being converted from the second format to the first format after  
performing a second operation of the series of operations;  
determining the origin of any operands of the series of operations being converted  
from the first format to the second format; and  
if the origin of any operand being converted from the first format to a second  
format is a conversion from the second format to the first format, then  
eliminating the redundant conversions from the second format to the first  
format and from the first format to the second format.~~  
expanding an application into a series of operations, the series of operations  
including a first operation and a second operation;  
examining the series of operations to determine that:  
a result of the first operation is an operand of the second operation, and  
the series of operations includes a first conversion of the result of the first  
operation from a first format to a second format after the first  
operation and a second conversion of the operand of the second

operation from the second format to the first format prior to the  
second operation; and  
eliminating the first conversion and the second conversion from the series of  
operations.

2. (Cancelled).
3. (Currently amended) The method of claim [[2]] 1, further comprising partially compressing the ~~remaining series of~~ operations after ~~redundant conversions of~~  
operands are eliminated the elimination of the first conversion and the second  
conversion.
4. (Currently amended) The method of claim 1, ~~wherein~~ further comprising  
determining that the series of operations includes a third conversion of the format  
of an operand of a result from the second format to the first format that is  
~~redundant~~ immediately followed by a fourth conversion of an operand from the  
first format to the second format in a first case and that is not ~~redundant~~  
immediately followed by the fourth conversion in a second case.
5. (Currently amended) The method of claim 4, wherein the third conversion ~~of the~~  
~~operand that is redundant in the first case and is not redundant in the second case~~  
~~occurs~~ and the fourth conversion occur within a loop, and ~~wherein the redundancy~~  
~~is eliminated at least in part by~~ and further comprising moving one or more  
~~operand conversions~~ the third conversion and the fourth conversion from inside  
the loop to outside the loop.

6. (Currently amended) The method of claim 1, wherein the first format is a ~~packed~~ unpacked floating point format and the second format is an ~~unpacked~~ packed floating point format.
7. (Currently amended) A method comprising:
- receiving an application, the application including one or more floating point calculations;
- expanding the application, the expansion of the application comprising [[by]]
- expanding the floating point calculations into a series of floating point operations on floating point operands, ~~the floating point operands being converted~~ the floating point operations including a conversion of an operand of a floating point operation from a packed format to an unpacked format prior to the performance of [[a]] the floating point operation ~~of the series of floating point operations~~, and converting the result of a floating point operation ~~of the series of floating point operations being converted~~ from the unpacked format to the packed format after the performance of the floating point operation;
- determining ~~the origin of the floating point operands~~ that a result of a first floating point operation of the series of floating point operation is an operand of a second floating point operation of the series of floating point operations;
- and
- ~~if the origin of any of the floating point operands is a conversion from the unpacked format to the packed format, then eliminating the redundant conversions of the operand from the unpacked format to the packed format~~

~~and from the packed format to the unpacked format~~ eliminating the conversion of the result of the first floating operation from the unpacked format to the packed format and the conversion of the operand of the second floating point operation from the packed format to the unpacked format.

8. (Cancel)
9. (Original) The method of claim 7, wherein the packed format is a standard floating point format.
10. (Original) The method of claim 9, wherein the standard floating point format is in accordance with Standard 754 of the Institute of Electrical and Electronic Engineers.
11. (Original) The method of claim 7, wherein the unpacked format is an operand format that may be used in performing floating point emulation of a floating point operation.
12. (Currently amended) The method of claim 11, wherein ~~converting~~ a conversion of a floating point operand from the packed format to the unpacked format comprises ~~converting~~ a conversion of the floating point operand to one or more integer values.
13. (Currently amended) The method of claim 7, further comprising partially compressing the remaining floating point operations, if any, after ~~redundant conversions of operands are eliminated~~ the elimination of the conversions.

14. (Currently amended) The method of claim 7, ~~wherein~~ further comprising  
determining that the series of floating point operations includes a third conversion  
~~of the format of a floating point operand of a result from the second format to the~~  
first format that is~~redundant~~ immediately followed by a fourth conversion of an  
operand from the first format to the second format in a first case and that is not  
~~redundant~~ immediately followed by the fourth conversion in a second case.
15. (Currently amended) The method of claim 14, wherein the first conversion ~~of the~~  
~~floating point operand that is redundant in the first case and is not redundant in~~  
~~the second case occurs~~ and the fourth conversion are within a loop, and ~~wherein~~  
~~the redundancy is eliminated at least in part by~~ the sequences of instructions  
include instructions that, when executed by a processor, cause the processor to  
perform operations comprising moving one or more operand conversions from  
inside the loop to outside the loop.
16. (Currently amended) A machine-readable medium having stored thereon data  
representing sequences of instructions that, when executed by a processor, cause  
the processor to perform operations comprising:  
~~receiving a series of operations, an operand of the series of operations being~~  
~~converted from a first format to a second format prior to performing a first~~  
~~operation of the series of operations, a result of the series of operations~~  
~~being converted from the second format to the first format after~~  
~~performing a second operation of the series of operations;~~  
~~determining the origin of any operands of the series of operations being converted~~  
~~from the first format to the second format; and~~

~~if the origin of any operand being converted from the first format to a second format is a conversion from the second format to the first format, then eliminating the redundant conversions from the second format to the first format and from the first format to the second format.~~

expanding an application into a series of operations, the series of operations

including a first operation and a second operation;

examining the series of operations to determine that:

a result of the first operation is an operand of the second operation, and

the series of operations includes a first conversion of the result of the first

operation from a first format to a second format after the first

operation and a second conversion of the operand of the second

operation from the second format to the first format prior to the

second operation; and

eliminating the first conversion and the second conversion from the series of

operations.

17. (Cancelled)

18. (Currently amended) The medium of claim ~~[[17]]~~ 16, wherein the sequences of instructions include instructions that, when executed by a processor, cause the processor to perform operations comprising partially compressing the ~~remaining~~ series of operations after redundant conversions of operands are eliminated the elimination of the first conversion and the second conversion.

19. (Currently amended) The medium of claim 16, wherein the sequences of instructions include instructions that, when executed by a processor, cause the processor to perform operations comprising determining that the series of operations includes a third conversion of the format of an operand of a result from the second format to the first format that is ~~redundant~~ immediately followed by a fourth conversion of an operand from the first format to the second format in a first case and that is not ~~redundant~~ immediately followed by the fourth conversion in a second case.
20. (Currently amended) The medium of claim 19, wherein the third conversion of the operand that is ~~redundant in the first case and is not redundant in the second case occurs~~ and the fourth conversion are within a loop, and wherein the ~~redundancy is eliminated at least in part by~~ the sequences of instructions include instructions that, when executed by a processor, cause the processor to perform operations comprising instructions for moving ~~one or more operand~~ the third and fourth conversions from inside the loop to outside the loop.
21. (Currently amended) The medium of claim 16, wherein the first format is a ~~packed~~ unpacked floating point format and the second format is an ~~unpacked~~ packed floating point format.
22. (Currently amended) A machine-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:

receiving an application, the application including one or more floating point calculations;

expanding the application, the expansion of the application comprising [[by]]

expanding the floating point calculations into a series of floating point operations on floating point operands, ~~the floating point operands being converted~~ the floating point operations including a conversion of an operand of a floating point operation from a packed format to an unpacked format prior to the performance of [[a]] the floating point operation ~~of the series of floating point operations~~, and converting the result of a floating point operation ~~of the series of floating point operations being converted~~ from the unpacked format to the packed format after the performance of the floating point operation;

determining ~~the origin of the floating point operands~~ that a result of a first floating point operation of the series of floating point operation is an operand of a second floating point operation of the series of floating point operations;

and

~~if the origin of any of the floating point operands is a conversion from the unpacked format to the packed format, then eliminating the redundant conversions of the operand from the unpacked format to the packed format and from the packed format to the unpacked format~~ eliminating the conversion of the result of the first floating operation from the unpacked format to the packed format and the conversion of the operand of the



second floating point operation from the packed format to the unpacked format.

23. (Cancelled)
24. (Original) The medium of claim 22, wherein the packed format is a standard floating point format.
25. (Original) The medium of claim 24, wherein the standard floating point format is in accordance with Standard 754 of the Institute of Electrical and Electronic Engineers.
26. (Original) The medium of claim 22, wherein the unpacked format is an operand format that may be used in performing floating point emulation of a floating point operation.
27. (Currently amended) The medium of claim 26, wherein ~~converting~~ a conversion of a floating point operand from the packed format to the unpacked format comprises ~~converting~~ a conversion of the floating point operand to one or more integer values.
28. (Currently amended) The medium of claim 22, wherein the sequences of instructions include instructions that, when executed by a processor, cause the processor to perform operations comprising partially compressing the remaining floating point operations, if any, after ~~redundant conversions of operands are eliminated~~ the elimination of the conversions.

29. (Currently amended) The medium of claim 22, wherein the series of instructions further comprises instructions for determining that the series of floating point operations includes a third conversion of the format of a floating point operand of a result from the second format to the first format that is redundant immediately followed by a fourth conversion of an operand from the first format to the second format in a first case and that is not redundant immediately followed by the fourth conversion in a second case.
30. (Currently amended) The medium of claim 29, wherein the third conversion of the floating point operand that is redundant in the first case and is not redundant in the second case occurs and the fourth conversion are within a loop, and wherein the redundancy is eliminated at least in part by moving the sequences of instructions include instructions that, when executed by a processor, cause the processor to perform operations comprising moving one or more operand the third and fourth conversions from inside the loop to outside the loop.